Date=30/09/2020
Lecture By=Arkesh Jaiswal
Subject ⇒Bootstrap

IN PREVIOUS LECTURE **(QUICK RECAP) Date-29/09/2020**	**In Today's Lecture (Overview)**
➢ Preprocessors ➢ SASS/SCSS ➢ Variables ➢ Nesting ➢ Mixins ➢ Operators ➢ Questions for the self Practice / CC for the Day	➢ Bootstrap ➢ Containers ➢ Grid system ➢ Tables ➢ Badges ➢ Links ➢ Buttons ➢ Forms ➢ MCQs ➢ Questions For Self Practice / Assignment For the Day

In today's Lecture We learned What bootstrap Is and how It works

Bootstrap

Bootstrap is the most popular CSS Framework for developing responsive and mobile-first websites.
Bootstrap 4 is the newest version of Bootstrap

19/03/17 Bootstrap 2

What is Bootstrap?

- Bootstrap is a free front-end framework (HTML and CSS) for faster and easier web development
- Bootstrap is famous for being developed with components that have the ability to follow the property of responsive designs
  - Responsive Design is about using CSS and HTML to resize, hide, shrink, enlarge, or move the content to make it look good on any screen
  - Responsive Design allow your page works for computer, tablets and mobile phones.

CSS

Copy-paste the stylesheet `<link>` into your `<head>` before all other stylesheets to load our CSS.

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min
.css"
integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGM
N5t9UJ0Z" crossorigin="anonymous">
```

Containers

Containers are the most basic layout element in Bootstrap and are required when using our default grid system. Containers are used to contain, pad, and (sometimes) center the content within them. While containers *can* be nested, most layouts do not require a nested container.

Bootstrap comes with three different containers:

- `.container`, which sets a `max-width` at each responsive breakpoint
- `.container-fluid`, which is `width: 100%` at all breakpoints
- `.container-{breakpoint}`, which is `width: 100%` until the specified breakpoint

The table below illustrates how each container's `max-width` compares to the original `.container` and `.container-fluid` across each breakpoint.

See them in action and compare them in our Grid example.

	Extra small <576px	**Small ≥576px**	**Medium ≥768px**	**Large ≥992px**	**Extra large ≥1200px**
`.container`	100%	540px	720px	960px	1140px
`.container-sm`	100%	540px	720px	960px	1140px

`.container-md`	100%	100%	720px	960px	1140px
`.container-lg`	100%	100%	100%	960px	1140px
`.container-xl`	100%	100%	100%	100%	1140px
`.container-fluid`	100%	100%	100%	100%	100%

Grid system

Use our powerful mobile-first flexbox grid to build layouts of all shapes and sizes thanks to a twelve column system, five default responsive tiers, Sass variables and mixins, and dozens of predefined classes.

How it works

Bootstrap’s grid system uses a series of containers, rows, and columns to layout and align content. It’s built with flexbox and is fully responsive. Below is an example and an in-depth look at how the grid comes together.

New to or unfamiliar with flexbox? Read this CSS Tricks flexbox guide for background, terminology, guidelines, and code snippets.

One of three columns
One of three columns
One of three columns

Copy

```
<div class="container">
  <div class="row">
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
  </div>
</div>
```

The above example creates three equal-width columns on small, medium, large, and extra large devices using our predefined grid classes. Those columns are centered in the page with the parent `.container`.

Breaking it down, here’s how it works:

- Containers provide a means to center and horizontally pad your site’s contents. Use `.container` for a responsive pixel width or `.container-fluid` for `width: 100%` across all viewport and device sizes.
- Rows are wrappers for columns. Each column has horizontal `padding` (called a gutter) for controlling the space between them. This `padding` is then counteracted on the rows with negative margins. This way, all the content in your columns is visually aligned down the left side.
- In a grid layout, content must be placed within columns and only columns may be immediate children of rows.
- Thanks to flexbox, grid columns without a specified `width` will automatically layout as equal width columns. For example, four instances of `.col-sm` will each automatically be 25% wide from the small breakpoint and up. See the auto-layout columns section for more examples.
- Column classes indicate the number of columns you’d like to use out of the possible 12 per row. So, if you want three equal-width columns across, you can use `.col-4`.
- Column `width`s are set in percentages, so they’re always fluid and sized relative to their parent element.

Grid options

While Bootstrap uses `em`s or `rem`s for defining most sizes, `px`s are used for grid breakpoints and container widths. This is because the viewport width is in pixels and does not change with the font size.

See how aspects of the Bootstrap grid system work across multiple devices with a handy table.

	Extra small <576px	**Small** ≥576px	**Medium** ≥768px	**Large** ≥992px	**Extra large** ≥1200px
Max container width	None (auto)	540px	720px	960px	1140px
Class prefix	`.col-`	`.col-sm-`	`.col-md-`	`.col-lg-`	`.col-xl-`
# of columns	12				
Gutter width	30px (15px on each side of a column)				
Nestable	Yes				
Column ordering	Yes				

Tables

Documentation and examples for opt-in styling of tables (given their prevalent use in JavaScript plugins) with Bootstrap.

Examples

Due to the widespread use of tables across third-party widgets like calendars and date pickers, we've designed our tables to be opt-in. Just add the base class `.table` to any `<table>`, then extend with custom styles or our various included modifier classes.

Using the most basic table markup, here's how `.table`-based tables look in Bootstrap. All table styles are inherited in Bootstrap 4, meaning any nested tables will be styled in the same manner as the parent.

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

```
<table class="table">
  <thead>
    <tr>
      <th scope="col">#</th>
      <th scope="col">First</th>
      <th scope="col">Last</th>
      <th scope="col">Handle</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>Mark</td>
      <td>Otto</td>
      <td>@mdo</td>
    </tr>
    <tr>
      <th scope="row">2</th>
      <td>Jacob</td>
      <td>Thornton</td>
      <td>@fat</td>
    </tr>
    <tr>
      <th scope="row">3</th>
      <td>Larry</td>
      <td>the Bird</td>
      <td>@twitter</td>
    </tr>
  </tbody>
</table>
```

You can also invert the colors—with light text on dark backgrounds—with `.table-dark`.

#	First	Last	Handle
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

Badges

Documentation and examples for badges, our small count and labeling component.

Example

Badges scale to match the size of the immediate parent element by using relative font sizing and `em` units.

Example heading **New**

Example heading **New**

Example heading **New**

Example heading **New**

Example heading **New**

Example heading **New**

```html
<h1>Example heading <span class="badge badge-secondary">New</span></h1>
<h2>Example heading <span class="badge badge-secondary">New</span></h2>
<h3>Example heading <span class="badge badge-secondary">New</span></h3>
<h4>Example heading <span class="badge badge-secondary">New</span></h4>
<h5>Example heading <span class="badge badge-secondary">New</span></h5>
<h6>Example heading <span class="badge badge-secondary">New</span></h6>
```

Badges can be used as part of links or buttons to provide a counter.

Notifications 4

```
<button type="button" class="btn btn-primary">
  Notifications <span class="badge badge-light">4</span>
</button>
```

Note that depending on how they are used, badges may be confusing for users of screen readers and similar assistive technologies. While the styling of badges provides a visual cue as to their purpose, these users will simply be presented with the content of the badge. Depending on the specific situation, these badges may seem like random additional words or numbers at the end of a sentence, link, or button.

Links

Using the contextual `.badge-*` classes on an `<a>` element quickly provide *actionable* badges with hover and focus states.

Primary Secondary Success Danger Warning Info Light Dark

```
<a href="#" class="badge badge-primary">Primary</a>

<a href="#" class="badge badge-secondary">Secondary</a>

<a href="#" class="badge badge-success">Success</a>

<a href="#" class="badge badge-danger">Danger</a>

<a href="#" class="badge badge-warning">Warning</a>

<a href="#" class="badge badge-info">Info</a>

<a href="#" class="badge badge-light">Light</a>

<a href="#" class="badge badge-dark">Dark</a>
```

Buttons

Use Bootstrap’s custom button styles for actions in forms, dialogs, and more with support for multiple sizes, states, and more

Examples

Bootstrap includes several predefined button styles, each serving its own semantic purpose, with a few extras thrown in for more control.

Primary Secondary Success Danger Warning Info Light Dark Link

```
<button type="button" class="btn btn-primary">Primary</button>

<button type="button" class="btn btn-secondary">Secondary</button>

<button type="button" class="btn btn-success">Success</button>

<button type="button" class="btn btn-danger">Danger</button>

<button type="button" class="btn btn-warning">Warning</button>

<button type="button" class="btn btn-info">Info</button>

<button type="button" class="btn btn-light">Light</button>

<button type="button" class="btn btn-dark">Dark</button>

<button type="button" class="btn btn-link">Link</button>
```

Conveying meaning to assistive technologies

Using color to add meaning only provides a visual indication, which will not be conveyed to users of assistive technologies – such as screen readers. Ensure that information denoted by the color is either obvious from the content itself (e.g. the visible text), or is included through alternative means, such as additional text hidden with the `.sr-only` class.

Disable text wrapping

If you don't want the button text to wrap, you can add the `.text-nowrap` class to the button. In Sass, you can set `$btn-white-space: nowrap` to disable text wrapping for each button.

Button tags

The `.btn` classes are designed to be used with the `<button>` element. However, you can also use these classes on `<a>` or `<input>` elements (though some browsers may apply a slightly different rendering).

When using button classes on `<a>` elements that are used to trigger in-page functionality (like collapsing content), rather than linking to new pages or sections within the current page, these links should be given a `role="button"` to appropriately convey their purpose to assistive technologies such as screen readers.

Link Button

```
<a class="btn btn-primary" href="#" role="button">Link</a>
```

```
<button class="btn btn-primary" type="submit">Button</button>
```

```
<input class="btn btn-primary" type="button" value="Input">
```

```
<input class="btn btn-primary" type="submit" value="Submit">
```

```
<input class="btn btn-primary" type="reset" value="Reset">
```

Outline buttons

In need of a button, but not the hefty background colors they bring? Replace the default modifier classes with the `.btn-outline-*` ones to remove all background images and colors on any button.

Primary Secondary Success Danger Warning Info Light Dark

```
<button type="button" class="btn btn-outline-primary">Primary</button>
```

```
<button type="button" class="btn btn-outline-secondary">Secondary</button>
```

```
<button type="button" class="btn btn-outline-success">Success</button>
```

```
<button type="button" class="btn btn-outline-danger">Danger</button>
```

```
<button type="button" class="btn btn-outline-warning">Warning</button>
```

```
<button type="button" class="btn btn-outline-info">Info</button>
```

```
<button type="button" class="btn btn-outline-light">Light</button>
```

```
<button type="button" class="btn btn-outline-dark">Dark</button>
```

Sizes

Fancy larger or smaller buttons? Add `.btn-lg` or `.btn-sm` for additional sizes.

Large button Large button

```
<button type="button" class="btn btn-primary btn-lg">Large button</button>
```

```
<button type="button" class="btn btn-secondary btn-lg">Large
button</button>
```

Small button Small button

```
<button type="button" class="btn btn-primary btn-sm">Small button</button>
```

```
<button type="button" class="btn btn-secondary btn-sm">Small
button</button>
```

Create block level buttons—those that span the full width of a parent—by adding `.btn-block`.

Block level button

Block level button

```
<button type="button" class="btn btn-primary btn-lg btn-block">Block level
button</button>
```

```
<button type="button" class="btn btn-secondary btn-lg btn-block">Block
level button</button>
```

Active state

Buttons will appear pressed (with a darker background, darker border, and inset shadow) when active. There's no need to add a class to `<button>`s as they use a pseudo-class. However, you can still force the same active appearance with `.active` (and include the `aria-pressed="true"` attribute) should you need to replicate the state programmatically.

Primary link Link

```
<a href="#" class="btn btn-primary btn-lg active" role="button"
aria-pressed="true">Primary link</a>
```

```
<a href="#" class="btn btn-secondary btn-lg active" role="button"
aria-pressed="true">Link</a>
```

Forms

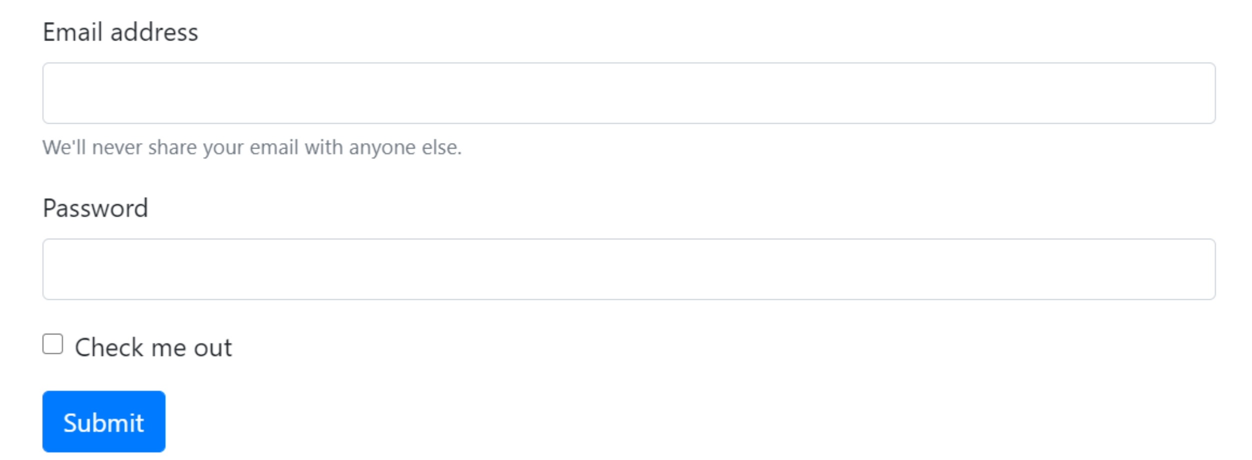
Examples and usage guidelines for form control styles, layout options, and custom components for creating a wide variety of forms.

Overview

Bootstrap's form controls expand on our Rebooted form styles with classes. Use these classes to opt into their customized displays for a more consistent rendering across browsers and devices.

Be sure to use an appropriate `type` attribute on all inputs (e.g., `email` for email address or `number` for numerical information) to take advantage of newer input controls like email verification, number selection, and more.

Here's a quick example to demonstrate Bootstrap's form styles. Keep reading for documentation on required classes, form layout, and more.



Email address

We'll never share your email with anyone else.

Password

□ Check me out

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1"
aria-describedby="emailHelp">
    <small id="emailHelp" class="form-text text-muted">We'll never share
your email with anyone else.</small>
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control"
id="exampleInputPassword1">
  </div>
  <div class="form-group form-check">
    <input type="checkbox" class="form-check-input" id="exampleCheck1">
    <label class="form-check-label" for="exampleCheck1">Check me
out</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

MCQs

1.How many units make up one row in bootstrap?

A = 15

B = 10

C = 12

D = 24

2.You cannot build responsive websites with bootstrap.

A = True

B = False

3.How do you include bootstrap in your project?

A = You have to install a package

B = Add a link to the bootstrap CDN in your html

4.What is bootstrap?

A = A CSS library

B = A JS library

C = A CSS preprocessor

Questions For Self Practice / Assignment For the Day

https://au-assignment.s3.amazonaws.com/assignment-fa219f89-02f3-4a73-a009-dadcc1daadb3.pdf